(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 505 797 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
09.02.2005 Bulletin 2005/06

(51) Int Cl.7: H04L 29/08

(21) Application number: 03291958.1

(22) Date of filing: 04.08.2003

(84) Designated Contracting States:
AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IT LI LU MC NL PT RO SE SI SK TR
Designated Extension States:
AL LT LV MK

(71) Applicant: ALCATEL
75008 Paris (FR)

(72) Inventors:
- Weik, Hartmut
70195 Stuttgart (DE)
- Rupp, Stephan, Dr.
74354 Besigheim (DE)

(74) Representative: Brose, Gerhard et al
Alcatel,
Intellectual Property Department Stuttgart
70430 Stuttgart (DE)

(54) A method, a communication network and a computer software product for distributing software packages or updates

(57) The invention relates to a method for distributing a software package or update over a communication network, where the communication network comprises a server system (S') and at least one client system (Ci'). The method comprises the steps of distributing (P3) the software package or update to the at least one client system (Ci') via the communication system (NCi') by the server system (S') and installing the software package or update on the at least one client system (Ci'), and comprises the further steps of distributing (P6) the software package or update to a further client system via the communication system (NCj') by the at least one client system. The invention relates also to a server system (S'), a client system (Ci'), a communication network (Nci', NCj'), and corresponding computer software products.

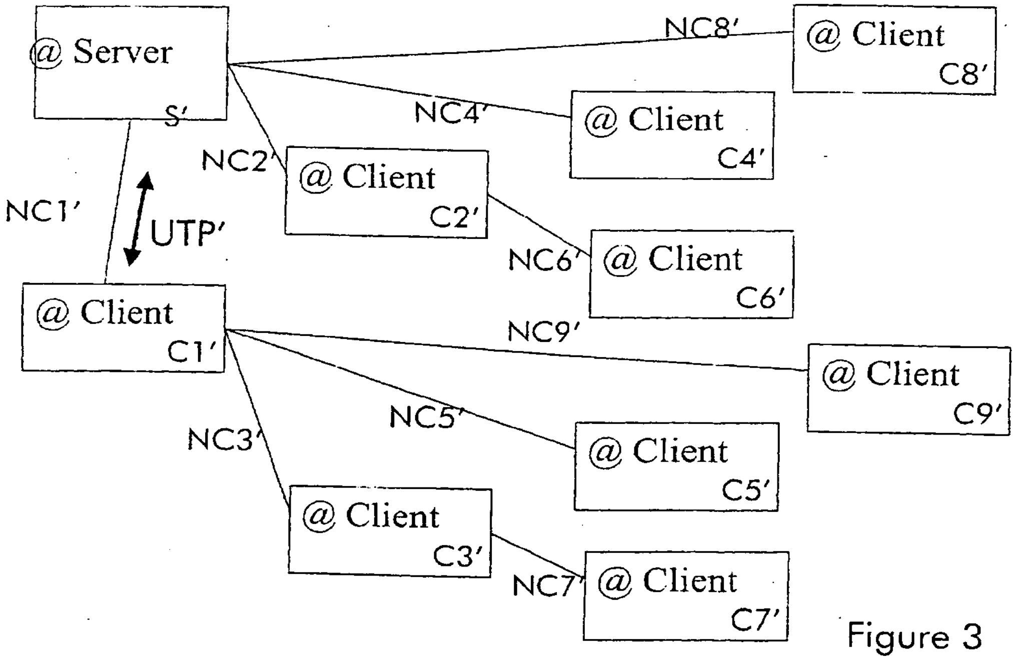


Figure 3

Printed by Jouve, 75001 PARIS (FR)

Description

[0001] The present invention relates to a method for distributing a software package or update over a communication network. The invention further relates to a communication network, a server system, a client system, and computer software products.

[0002] Due to the complexity of computer systems and (tele)communication systems, as well as the (tele) communication networks and the emerging techniques and developments in intruding systems, it is highly necessary to keep these systems up-to-date, i.e. keeping the software operable on its latest release.

[0003] There are many techniques known for keeping software driven systems up-to-date, e.g. manually or automatically patching, new (re-)installations, or updates. Especially for virus protection means virus patterns and treatments are deployed continuously in order to enable such a software driven system recognizing infections and applying the corresponding treatment.

[0004] Systems and methods for distributing software (applications and data) to many clients over a network are well known. Usually there are servers for deploying the software updates and clients that consume these software updates. There exist already many variants of update (transfer) protocols. One variant is that the server continuously updates the client's software.

[0005] Another variant is that the client is more active and requests for software updates, e.g. event-driven.

[0006] The US Patent Application No. 6,123,737 describes an update (transfer) protocol for deploying a software package by triggers that are sent to servers. In response the servers create a notification package for a client. The notification instructs the server to automatically push a software package to the client computer over a communications interface.

[0007] A system comprising self-updating clients, realized by a managed update procedure using a network connection to a supporting server is known from the US Patent Application No. 6,067,351.

[0008] An example of a self-distributing piece of software is a worm, e.g. the Code Red virus. This virus was one of the first of a family of new self-propagating malicious codes that exploits network systems. The Code Red worm is self-replicating malicious code that exploits a vulnerability in several servers. A worm attack proceeds as follows. The virus attempts to connect to a randomly chosen host assuming that a web server will be found. Upon a successful connection the attacking host sends a crafted HTTP GET request to the victim, attempting to exploit a buffer overflow in an indexing service. The same exploit (HTTP GET request) is sent to each of the randomly chosen hosts due to the self-propagating nature of the worm.

[0009] Depending on the configuration of the host which receives this request, there are varied consequences, e.g. when the exploit is successful, the worm begins executing on the victim host. In addition to possible web site defacement, infected systems may experience performance degradation as a result of the scanning activity of this worm. This degradation can become quite severe since it is possible for a worm to infect a machine multiple times simultaneously. Non-compromised systems and networks that are being scanned by other infected hosts may experience severe denial of service. Furthermore, it is important to note that while the Code Red worm appears to merely deface web pages on affected systems and attack other systems, the indexing vulnerability it exploits can be used to execute arbitrary code in the local system security context. This level of privilege effectively gives an attacker complete control of the victim system.

[0010] Due to the exponential distribution behavior of such virus infections and propagating (network) malfunctions there is a need for a fast and efficient remedy (cure).

[0011] This problem is solved for distributing a software package or update over a communication network, the communication network comprises a server system and at least two client systems, said method comprises the steps of:

- distributing the software package or update to the at least one client of the at least two client systems via the communication system by the server system and
- distributing the software package or update (recursively) to a further client system via the communication system by the at least one client of the at least two client systems (until the further client system is already updated).

[0012] The problem is further solved by a communication network comprising a server system and at least one client system, the server system comprising distribution means for distributing a software package or update to the at least one client system, the at least one client system comprises installation means for installing the software package or update on the at least one client system, where the at least one client system comprises distribution means for distributing the software package or update to a further client system, too.

[0013] Accordingly, the problem is solved inter alia by a server system for a communication network comprising at least one client system, the server system comprising distribution means for distributing a software package or update to the at least one client system, the at least one client system comprises installation means for installing the software package or update on the at least one client system, where the server system further comprises control means for controlling the at least one client to distribute the software package or update to a further client system.

[0014] And the problem is solved by a client system for a communication network comprising a server system, the server system comprising distribution means for distributing a software package or update to the client system, the client system comprises installation means for installing the software package or update on the client system, where the client system comprises distribution means for distributing the software package or update to a further client system.

[0015] The problem is solved by a computer software product realizing a software package or update to be distributed over a communication network to a client system, the computer software product comprising programming means implementing deployment means and container means for distributing the software package or update to a further client system (recursively) via a communication system.

[0016] And the problem is solved by a computer software product for distributing a software package or update over a communication network as described in the above method.

[0017] In other words a patch or update deployment pattern itself acts like a virus, infecting all systems that are not vaccinated with the method the vaccination should prevent. After being infected, the system is forced to distribute the remedy virus. In a subsequent step the virus patches the system in a way that e.g. viruses, using this method of access and the remedy itself are not able to infect a cured system again.

[0018] The effect of this procedure is, that all systems, that are not cured will help to distributed the remedy. This will result in a very quick distribution of the required patches.

[0019] Accordingly, it is an advantage of the present invention to provide fast and effective distribution of software patches and updates in a communication network.

[0020] Another advantage of the present invention is the increased security and reliability.

[0021] A further advantage of the present invention is the silent installation of patches that enhance update quality and patch quality thus indirectly reducing the requirements on activity of system operators.

[0022] Yet another advantage of the present invention is that the invention provides a method with an advanced deployment pattern that can even cope with worms and communication network degradations.

[0023] These and many other objects and advantages of the present invention will become apparent to those of ordinary skill in the art from a consideration of the drawings and ensuing description.

Figure. 1 is a schematic drawing of a prior art deployment pattern of an update.

Figure. 2 is a schematic drawing of a method for distributing a software package or update over a communication network according to the invention.

Figure. 3 is a schematic drawing deployment pattern of an update forced by the method according to the invention.

[0024] Figure 1 shows a server system S and a set of client systems C1, C2, ..., C9. Each client system is connected via a network connection NC1, NC2, ..., NC9 with the server system S, respectively. The server system S and a client system Ci communicates by an update transfer protocol UTP over the network connection NCi.

[0025] Thus the server S can update the client system's Ci's software or the client system Ci could update its software by commonly identifying the corresponding software package or update and downloading it from the server system S and installing it on the client system Ci using the update transfer protocol UTP.

[0026] There are 9 client systems C1, C2, ..., C9 shown. When a new update arises, the server system S has to process 9 updates, one for each client system C1, C2, ..., C9 in order to update all the client systems C1, C2, ..., C9. This requires about 9 times of one update. In general n client updates would have a time complexity of O(n).

[0027] Figure 2 illustrates the steps of the distributing method according to the invention and where, i.e. at which site, these steps have to be performed. The figure shows a server system site S', a network connection site NCi', and a client system site Ci'. The figure further shows update process phases, namely a new software package is available P1, an encapsulation in a virus shell P2, a distribution phase P3, an infection phase P4, an installation of the software package P5, and a further distribution phase P6.

[0028] The new software package is available P1 at the server system site S' initiates the process. There, at the server system site S', the new software package becomes a virus by the encapsulation in a virus shell P2. The result is deployed via the network connection site NCi', received at the client system site Ci' while the distribution phase P3. The client system site Ci' becomes infected while the infection phase P4, and the encapsulated software is installed while the installation of the software package P5. Then, in advance the virus is further deployed over another network connection NCj' in the further distribution phase P6.

[0029] In other words: deploy updates by generating a virus comprising deployment means and container means for said software package and distributing said virus over said communication network by a server system, and infecting said at least one client system and forcing said client system further installing said software package and distributing said virus over said communication network for infecting further client systems.

[0030] The client itself might have the deployment means to propagate update information. An advanced update transfer protocol might enable a client system to provide feedback about the installation and the propagation.

[0031] The method formalizes the provision of a system to distribute patches, e.g. against viruses, using the virus' distribution mechanism. The system might invoke operators to indicate the remedy (available update) of the system including the ability e.g. to provide charging for or to control the distribution.

[0032] Figure 3 shows a (advanced) server system S' and a set of (advanced) client systems C1', C2', ..., C9'. The server system S' and the client systems C1', C2', ..., C9' are inter-connected via the network connections NC1', NC2', ..., NC9'.

[0033] The server can distribute software updates according to the method illustrated in figure 2. There are 9 client systems C1', C2', ..., C9' shown. When a new update arises, the new update is deployed in waves.

[0034] Assume a first deployment from the server system S' to the client system C1' requiring the time of one update. In the second deployment wave the server system S' and the client C1' deploy respectively the update to two further client system C2' and C3', respectively, via the network connections NC2' and NC3'. In the third deployment wave the server system S' and the already updated client systems C1', C2', and C3' deploy respectively the update to further 4 client systems C4', C5', C6', and C7', respectively, via the network connections NC4', NC5', NC6, and NC7. In a further deployment wave the remaining client systems C8' and C9' are updated via the network connections NC8' and NC9'. The whole procedure requires about 4 times of one update. In general n client updates would have a time complexity of O(log n). The effect of the claimed method is that all systems, that are cured will help to distribute the remedy. This will result in a very quick distribution of the required patches for the operating systems.

[0035] In order to highly multiple updates the advanced update transfer protocol might comprise means for providing feedback on an update, e.g. which further clients were also updated, recursively. Such an information could be used at the advanced server system keeping track of the update deployments. The coordination of the updates might be randomly driven, self-organizing, in a dynamic way based on environmental aspects like network connectivity, or even static, i.e. the deployment graph (tree) is fix.

[0036] The virus remedy works using a simple principle. It is itself a virus, that infects all client systems that are not vaccinated with the method the vaccination should prevent. After being infected, the client system is forced to distribute the remedy virus.

[0037] In a subsequent step the virus patches the client system in a way that viruses, using this method of access and the remedy itself are not able to infect a cured system again.

[0038] An advanced update transfer protocol might have capabilities interactively to aggregate and coordinate update resources, e.g. for managing multiple client updates, partial updates, or even an assignment about update responsibility or update authority.

[0039] The software package or update itself could be designed to comprise the virus functionality, i.e. a virus shell.

[0040] Currently there is a trend in computer science to solve problems using nature-analogous methods, e. g. neuronal networks, genetic algorithms etc. The corresponding biological object to this invention is a retrovirus.

[0041] Retroviruses are infectious particles consisting of an RNA genome (the software update) packaged in a protein capsid, surrounded by a lipid envelope (the container). This lipid envelope contains polypeptide chains including receptor binding proteins which link to the membrane receptors of the host cell, initiating the process of infection (the distribution).

[0042] Retroviruses contain RNA as the hereditary material in place of the more common DNA. In addition to RNA, retrovirus particles also contain the enzyme reverse transcriptase (or RTase), which causes synthesis of a complementary DNA molecule (cDNA) using virus RNA as a template (the update).

[0043] When a retrovirus infects a cell, it injects its RNA into the cytoplasm of that cell along with the reverse transcriptase enzyme. The cDNA produced from the RNA template contains the virally derived genetic instructions and allows infection of the host cell to proceed (the recursive distribution).

[0044] The capsis could e.g. preferably realized by an mobile agent using a mobile agent platform or any other applicable technique like the security leaks in several web servers that are e.g. used by Code Red.

Claims

1. A method for distributing a software package or update over a communication network, the communication network comprises a server system (S') and at least two client systems (Ci', Cj'), said method comprises the step of:

   - distributing (P3) the software package or update to the at least one client (Ci') of the at least two client systems via the communication system (NCi') by the server system (S'),

   characterized by comprising the further step of

   - distributing (P6) the software package or update to a further client system (Cj') via the communication system (NCj') by the at least one client (Ci') of the at least two client systems.

2. The method according to claim 1, **characterized in that** said method comprises a further step installing (P5) the software package or update on the at least one (Ci') of the at least two client systems.

3. The method according to claim 1, **characterized in that** said software package is an anti-virus pattern.

4. The method according claim 3, **characterized in that** the method comprising the step of removing the virus intruding leak.

5. The method according to claim 1, **characterized by** comprising a further step of informing the at least one (Ci') of the at least two client systems about the distributing activity.

6. The method according to claim 1, **characterized by** comprising a further step of informing the server system (S') about the installation or distribution activity of the at least one client system.

7. A communication network comprising a server system (S') and at least one client system (Ci'), the server system (S') comprising distribution means for distributing a software package or update to the at least one client system (Ci'), the at least one client system (Ci') comprises installation means for installing the software package or update on the at least one client system (Ci'), **characterized in that** the at least one client system (Ci') comprises distribution means for distributing the software package or update to a further client system (Cj').

8. The communication network according to claim 5, **characterized in that** the server system (S') further comprises control means for controlling the at least one client system (Ci') to distribute the software package or update to a further client system (Cj').

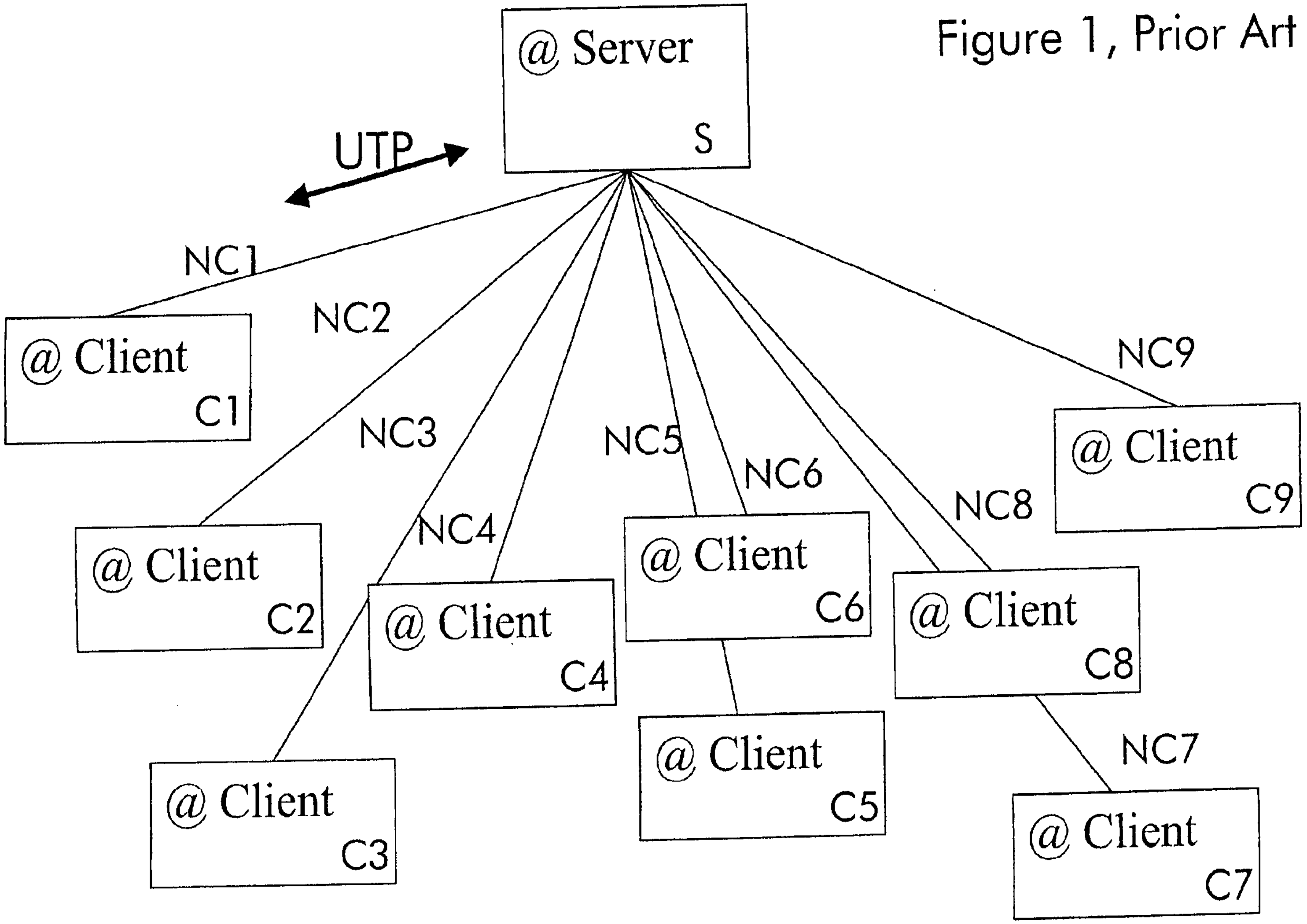
9. A server system (S') for a communication network comprising at least one client system (Ci'), the server system (S') comprising distribution means for distributing a software package or update to the at least one client system, the at least one client system (Ci') comprises installation means for installing the software package or update on the at least one client system (Ci'), **characterized in that** the server system (S') further comprises control means for controlling the at least one client system (Ci') to distribute the software package or update to a further client system (Cj').

10. A client system (Ci') for a communication network comprising a server system (S'), the server system (S') comprising distribution means for distributing a software package or update to a client system (Ci'), the client system (Ci') comprises installation means for installing the software package or update on the client system (Ci'), **characterized in that** the client system (Ci') comprises distribution means for distributing the software package or update to a further client system (Cj').

11. A computer software product realizing a software package or update to be distributed over a communication network to a client system (Ci'), **characterized by** comprising programming means implementing deployment means and container means for distributing (P6) the software package or update to a further client system (Cj') via a communication system (NCj').

12. A computer software product for distributing a software package or update over a communication network, **characterized by** comprising programming means implementing the method according to claim 1.

Figure 1, Prior Art



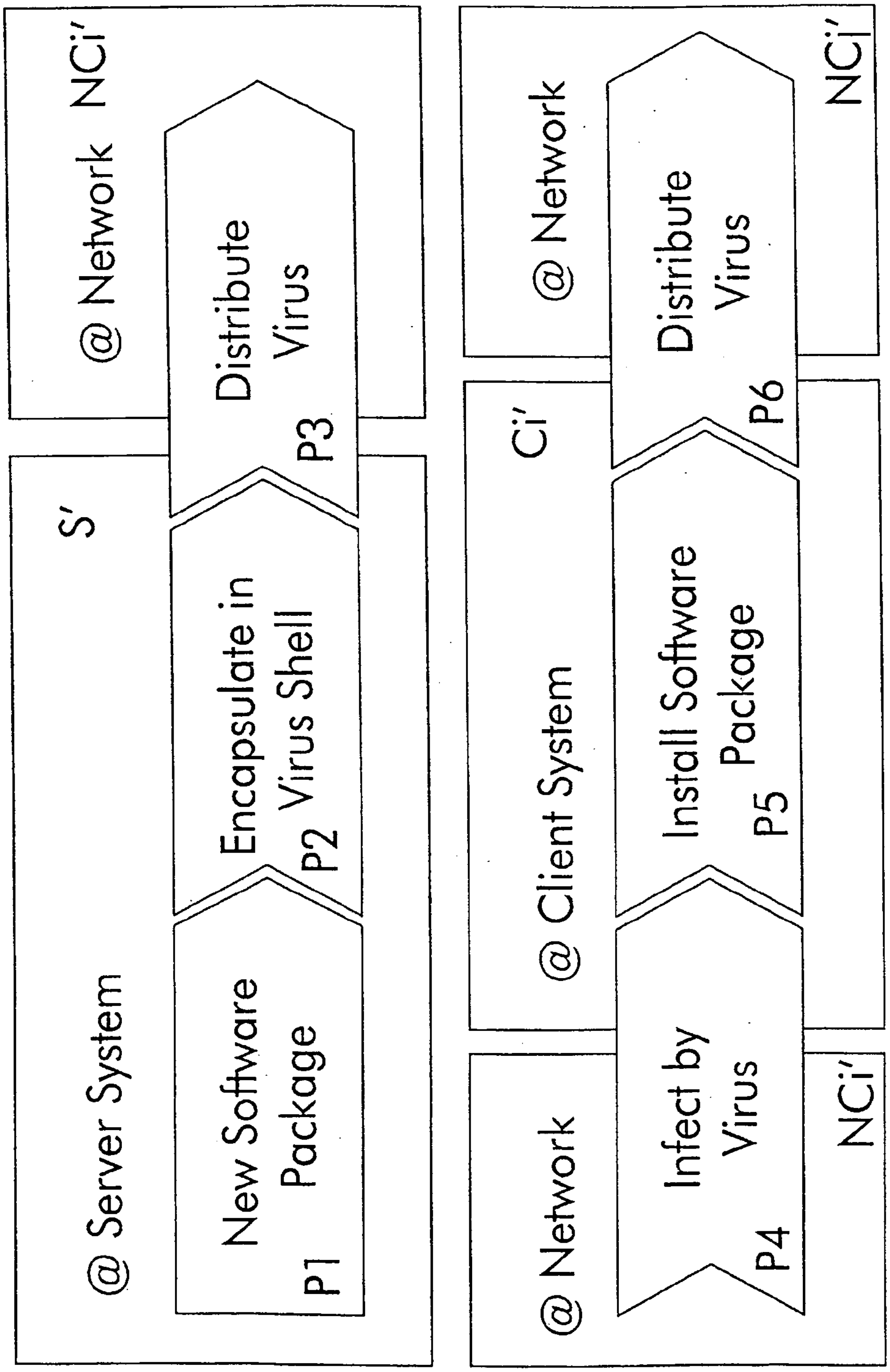

@ Server System
S'
New Software Package
P1
Encapsulate in Virus Shell
P2
Distribute Virus
P3
@ Network NCi'
@ Network
NCi'
Infect by Virus
P4
@ Client System
Ci'
Install Software Package
P5
Distribute Virus
P6
@ Network
NCi'

Figure 2

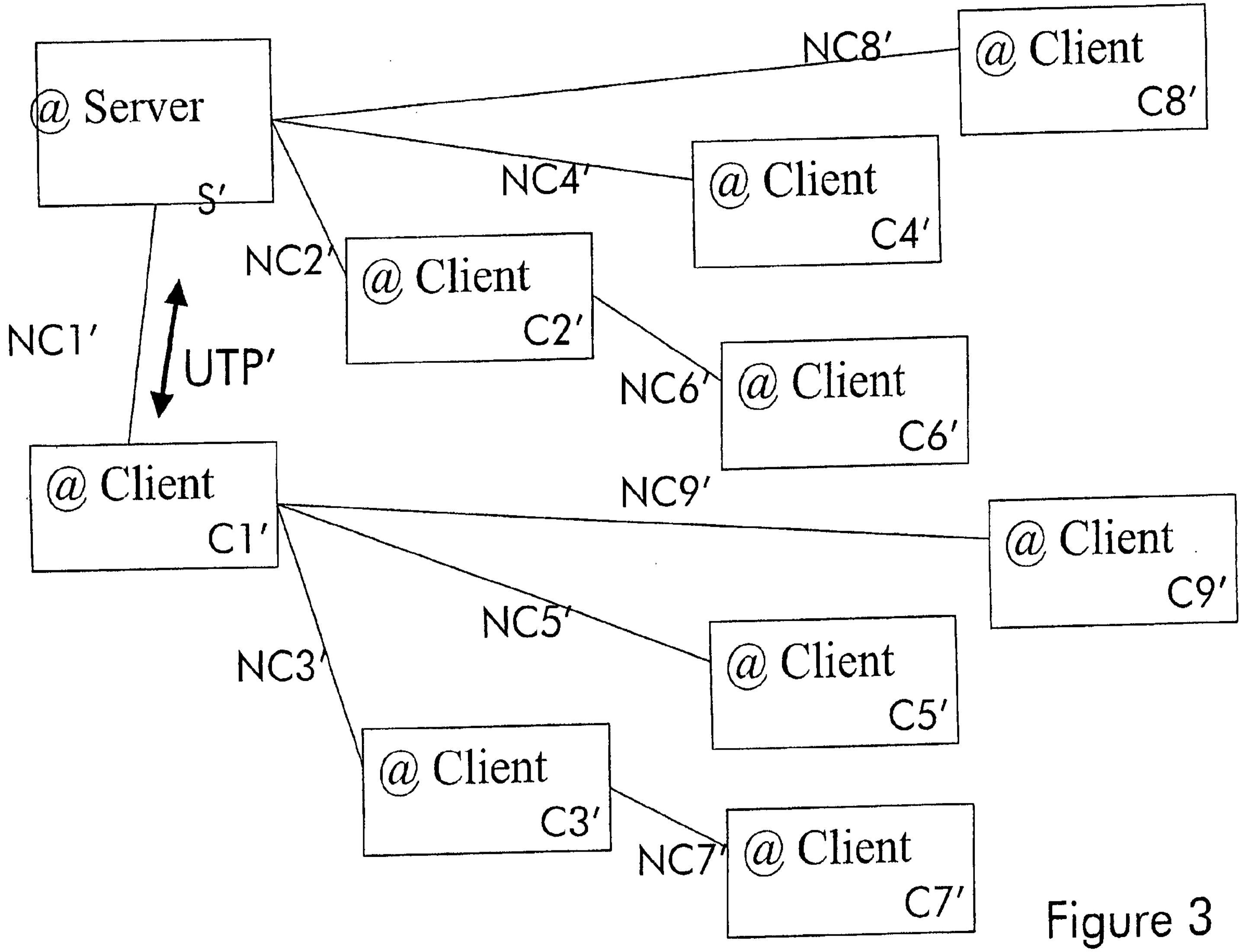

@ Server
S'
NC1'
UTP'
@ Client
C1'
NC2'
@ Client
C2'
NC4'
@ Client
C4'
NC8'
@ Client
C8'
NC6'
@ Client
C6'
NC9'
@ Client
C9'
NC5'
@ Client
C5'
NC3'
@ Client
C3'
NC7'
@ Client
C7'

Figure 3

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 03 29 1958

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
X	WO 02 29551 A (CORDOVA ROIS ;INTEL CORP (US)) 11 April 2002 (2002-04-11) * page 2, line 8 - page 5, line 30 * * figures 1-3 * ---	1-12	H04L29/08
X	US 6 052 721 A (SHEPHERD BRUCE ET AL) 18 April 2000 (2000-04-18) * column 1, line 34 - line 51 * * column 2, line 45 - column 4, line 30 * * figures 3-5 * ---	1-12	
A	BROOKS R. ET AL: "A Model for Mobile Code Using Interacting Automata" IEEE TRANS. ON MOBILE COMPUTING, vol. 1, no. 4, October 2002 (2002-10) - December 2002 (2002-12), pages 313-326, XP002262344 * abstract * * Section 4 on pages 323-324 * -----	1-12	
			TECHNICAL FIELDS SEARCHED (Int.Cl.7)
			H04L G06F

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
MUNICH	21 November 2003	Homan, P

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 03 29 1958

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report.
The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

21-11-2003

Patent document cited in search report		Publication date	Patent family member(s)			Publication date
WO 0229551	A	11-04-2002	AU	9499001	A	15-04-2002
			CA	2423722	A1	11-04-2002
			DE	10196732	T0	04-09-2003
			FI	20030509	A	03-04-2003
			GB	2383869	A	09-07-2003
			NO	20031512	A	03-06-2003
			WO	0229551	A2	11-04-2002
US 6052721	A	18-04-2000	US	6560643	B1	06-05-2003
			EP	0689325	A2	27-12-1995
			JP	8083245	A	26-03-1996
			ZA	9504146	A	06-03-1996

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82